


```

MM      MM      TTTTTTTTTT  HH      HH      RRRRRRRR      AAAAAA      NN      NN      DDDDDDDD      000000      MM      MM
MM      MM      TTTTTTTTTT  HH      HH      RRRRRRRR      AAAAAA      NN      NN      DDDDDDDD      000000      MM      MM
MMM     MMM     TT          HH      HH      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MMM     MMM
MMM     MMM     TT          HH      HH      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MMM     MMM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NNNN     NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NNNN     NN      DD      DD      00      00      MM      MM
MM      MM      TT          HHHHHHHHHH  RRRRRRRR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HHHHHHHHHH  RRRRRRRR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AAAAAAAAAA  NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AAAAAAAAAA  NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NN      NN      DDDDDDDD  000000  MM      MM
MM      MM      TT          HH      HH      RR      RR      AA      AA      NN      NN      DDDDDDDD  000000  MM      MM

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

(2)	57	DECLARATIONS
(3)	84	MTH\$RANDOM - CALLable function

```

0000 1      .TITLE MTH$RANDOM - General Purpose Random Number Generator
0000 2      .IDENT /1-006/ ; File: MTHRANDOM.MAR
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 *  ALL RIGHTS RESERVED.
0000 11 *
0000 12 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 *  TRANSFERRED.
0000 18 *
0000 19 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 *  CORPORATION.
0000 22 *
0000 23 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: Mathematics Library
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 This is a general random number generator. It is
0000 35 of the multiplicative congruential type, and hence
0000 36 is fast, although prone to certain classes of
0000 37 non-random sequences.
0000 38
0000 39 ENVIRONMENT: any access mode, normally user mode
0000 40 modular, AST reentrant procedure
0000 41 --
0000 42 +
0000 43 AUTHOR: Peter F. Conklin, CREATION DATE: 28-Dec-77
0000 44
0000 45 MODIFIED BY:
0000 46
0000 47 Peter F. Conklin, 1-Feb-78: VERSION 01
0000 48 01 - Original creation.
0000 49 02 - Change to LIB$ rather than MTH$.
0000 50 03 - Change back to MTH$. Some more comments. JMT 4-Jan-78
0000 51 04 - Correct documentation on usage.
0000 52 1-005 - Put version number in standard format (three digits in
0000 53 edit number field). JBS 16-NOV-78
0000 54 1-006 - Add "-" to the PSECT directive. JBS 22-DEC-78
0000 55 :-

```



```

0000 57      .SBTTL  DECLARATIONS
0000 58
0000 59 ::
0000 60      INCLUDE FILES:
0000 61 ::
0000 62      NONE
0000 63 ::
0000 64      MACROS:
0000 65 ::
0000 66      NONE
0000 67 ::
0000 68      EQUATED SYMBOLS:
0000 69 ::
0000 70
00000004 0000 71      SEED=4                      ; Only argument is the seed
0000 72
0000 73 ::
0000 74      OWN STORAGE:
0000 75 ::
0000 76      NONE
0000 77 ::
0000 78
0000 79 ::
0000 80      PSECT DECLARATIONS:
0000 81 ::
00000000 0000 82      .PSECT  _MTH$CODE,PIC,SHR,NOWRT,LONG,EXE

```

```
0000 84 .SBTTL MTH$RANDOM - CALLable function
0000 85 :++
0000 86 : FUNCTIONAL DESCRIPTION:
0000 87 :
0000 88 : RANDOM - General Purpose Pseudo Random Number Generator
0000 89 :
0000 90 : Call the function MTH$RANDOM (seed) to obtain the
0000 91 : next pseudo-random number. The seed is updated by
0000 92 : the function automatically as a side effect. The
0000 93 : result is a floating point number that is uniformly
0000 94 : distributed in the range 0.0 inclusive to 1.0 exclusive.
0000 95 : There are no restrictions on the seed, although
0000 96 : it should be initialized to different values on
0000 97 : separate runs.
0000 98 :
0000 99 : The algorithm used is to update the seed as:
0000 100 :
0000 101 :     seed = 69069. * seed + 1      (mod 2**32)
0000 102 :
0000 103 : and then to convert the seed to floating point.
0000 104 :
0000 105 : Note, because the result is never 1.0, a simple
0000 106 : way to get a uniform random integer selector is
0000 107 : to multiply by the number of cases. For example
0000 108 : if a uniform choice among 5 situations is to be
0000 109 : made, then the following FORTRAN statement will
0000 110 : work:
0000 111 :     GO TO (1,2,3,4,5) 1+IFIX(5.*MTH$RANDOM(seed))
0000 112 :
0000 113 : Note that the explicit IFIX is necessary before
0000 114 : adding 1 in order to avoid a possible rounding
0000 115 : during the normalization after the floating add.
0000 116 :
0000 117 : This is a general random number generator. It is
0000 118 : of the multiplicative congruential type, and hence
0000 119 : is fast, although prone to certain classes of
0000 120 : non-random sequences. This non-random behavior
0000 121 : typically arises when considering triples of
0000 122 : numbers generated by this method.
0000 123 :
0000 124 : For more information on congruential generators,
0000 125 : see:
0000 126 :     Random Number Generation (pp. 1192-1197)
0000 127 :     by G. Marsaglia
0000 128 :
0000 129 : in:  Encyclopedia of Computer Science
0000 130 :     edited by Anthony Ralston
0000 131 :     Petrocelli (New York, 1976)
0000 132 :
0000 133 : CALLING SEQUENCE:
0000 134 :
0000 135 :     result.wf.v = MTH$RANDOM (seed.mlu.r)
```

```

0000 137 :
0000 138 : INPUT PARAMETERS:
0000 139 :
0000 140 :     SEED.mlu.r           Longword seed is modified each call
0000 141 :
0000 142 : IMPLICIT INPUTS:
0000 143 :
0000 144 :     NONE
0000 145 :
0000 146 : OUTPUT PARAMETERS:
0000 147 :
0000 148 :     NONE
0000 149 :
0000 150 : IMPLICIT OUTPUTS:
0000 151 :
0000 152 :     NONE
0000 153 :
0000 154 : FUNCTION VALUE
0000 155 :
0000 156 :     Returns in R0 a single-precision floating point value between
0000 157 :     0.0 inclusive and 1.0 exclusive.
0000 158 :
0000 159 : SIDE EFFECTS:
0000 160 :
0000 161 :     NONE
0000 162 :
0000 163 :--

```



```
0000 0000 165 .ENTRY MTH$RANDOM,0 ;no registers save, clear IV
0002 166
0002 167 :-
0002 168 : If this were to be placed as an inline expansion, then
0002 169 : EMUL SEED,#69069,#1,R0 should replace the next two
0002 170 : instructions because this would prevent the possibility
0002 171 : of integer overflow trapping.
0002 172 :-
0002 173
04 BC 00010DCD 8F C4 0002 174 MULL2 #69069,@SEED(AP) ;update seed with multiplier
04 BC 04 BC D6 000A 175 INCL @SEED(AP) ;increment seed to protect
000D 176 ; against strange seeds
000D 177
000D 178 :
000D 179 : The next instructions convert the seed from unsigned integer
000D 180 : to floating point in the range 0.0 to 1.0 exclusive.
000D 181 :
000D 182 :
50 04 BC 18 08 EF 000D 183 EXTZV #8,#24,@SEED(AP),R0 ;Get the most significant bits
0013 184 ; of the seed in the range
0013 185 ; 0 .. (2**24)-1
50 50 4E 0013 186 CVTLF R0,R0 ;Convert to floating without
0016 187 ; rounding. The result is
0016 188 ; positive and in the range
0016 189 ; 0.0 .. (2.0**24)-1.0
0016 190
0016 191 :
0016 192 : If this were to be placed as an inline expansion, then
0016 193 : MULF #^X00003480,R0 could replace the next two instructions.
0016 194 :
0016 195 :
50 0C00 05 13 0016 196 BEQL 10$ ;If zero, already correct
A2 0018 197 SUBW #24@7,R0 ;DIVF #^F2.0**24
001D 198 ; the result is now in the
001D 199 ; range 0.0 .. 1.0 exclusive
001D 200
04 001D 201 10$: RET
001E 202
001E 203 .END
```


MTH\$RANDOM
Symbol table

F 9
- General Purpose Random Number Generator 16-SEP-1984 01:48:11 VAX/VMS Macro V04-00 Page 6
6-SEP-1984 11:26:48 [MTHRTL.SRC]MTHRANDOM.MAR;1 (5)

MTH\$RANDOM 00000000 RG 01
SEED = 00000004

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR
MTH\$CODE	0000001E (30.)	01 (1.)	PIC USR
			CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
			CON REL LCL SHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.08	00:00:00.55
Command processing	134	00:00:00.53	00:00:04.12
Pass 1	71	00:00:00.46	00:00:01.67
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.42	00:00:01.53
Symbol table output	2	00:00:00.00	00:00:00.19
Psect synopsis output	2	00:00:00.04	00:00:00.08
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	291	00:00:01.55	00:00:08.22

The working set limit was 900 pages.
1893 bytes (4 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 2 non-local and 1 local symbols.
203 source lines were read in Pass 1, producing 10 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:MTHRANDOM/OBJ=OBJ\$:MTHRANDOM MSRC\$:MTHRANDOM/UPDATE=(ENH\$:MTHRANDOM)

0263

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY